REMARKS

This reply is filed in response to the Final Office Action mailed April 18, 2007, rejecting claims 1 – 63 under 35 U.S.C. § 112 and 35 U.S.C. § 103. In view of the amendments above and the remarks that follow, the Applicants submit that all pending claims are in condition for allowance.

I. Oath/Declaration

This application is a continuation of United States Patent Application Serial Number 10/449,732, filed May 30, 2003, entitled "Virtual Processor Methods and Apparatus with Unified Event Notification and Consumer-Producer Memory Operations." A copy of the Declaration filed from that parent application is filed herewith.

II. The Claims are in Full Compliance with 35 U.S.C. § 112

Claims 36, 48 and 62 stand rejected under 35 U.S.C. § 112, second paragraph, as allegedly indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicants regards as the invention. The claims are amended as suggested by the Examiner, namely, to recite, processing an event by the thread to which "the event" is delivered.

In view of these amendments, the Applicants request that all 35 U.S.C. § 112 rejections be withdrawn.

III. Rejection of Claims 1 – 3, 5 – 14, 16 – 23, 27 – 37, 39 – 52 and 56 – 63 Under 35 U.S.C. § 103

The captioned claims stand rejected as allegedly unpatentable over Brown, III et al, US Patent 6,240,508 ("Brown") in view of Jagannathan et al, US Patent 5,692,193("Jagannathan").

Independent Claim 1

Claim 1 is directed towards an embedded processor comprising a plurality of processing units that each execute processes or threads (collectively, "threads"). One or more execution units are shared by the processing units and execute instructions from the threads. An event delivery mechanism is in communications coupling with the plurality of processing units and delivers events (e.g., interrupts) to respective threads without execution of instructions by the processing units.

Brown or Jagannathan, both individually and in combination, fail to teach or suggest an embedded processor meeting the limitations of claim 1. The record, moreover, is devoid of any evidence suggesting the propriety of combining those references.

For example, Brown purports to disclose a synchronized macropipelined microprocessor chip and a CPU having several distinct "units," including an "execution unit." *See,* Brown, col 7, lines 24 – 48. Nowhere does the publication teach or suggest an event delivery mechanism that is in communication with a plurality of processing units and that delivers events to respective threads without execution of instructions by the processing units. The Examiner does not contend otherwise — but, instead, asserts that Jagannathan remedies this deficiency.

In this regard, the Examiner cites Jagannathan, at col. 25, lines 22 – 26, arguing that it discloses delivering events without executing instructions. *See* Office Action mailed April 18, 2007, pp. 3, ¶ 1. The Applicants respectfully disagree with that interpretation of the cited text, believing that the Examiner may have mistakenly read it out of context.

The sentence of Jagannathan relied on by the Examiner provides that "[s]ending a thread a signal is equivalent to interrupting the thread and pushing a continuation containing the signal handler and its arguments onto the thread's stack, and resuming the thread which causes the

signal handler to be executed." *See* Jagannathan, col. 25, lines 22 – 26. However, the surrounding text shows that in order to effect delivery of exceptions, the reference necessarily requires execution of processor instructions. Jagannathan clearly details the operation of the exception dispatcher, even going so far as to include pseudo-code for the dispatcher.

```
1:    (define (exception-dispatcher type . args)
2:        (save-current-continuation)
3:        (let ((target handler (get-target&handler type args)))
4:            (cond ((eq? target (current-thread))
5:                (apply handler args))
6:                (else
7:                (signal target handler args)
8:                (case ((exception-priority type))
9:                ((continue) (return))
10:               ((immediate) (switch-to-thread target))
11:               ((reschedule) (yield-processsor)))))))
```

Jagannathan, at col 24, line 24, to col 25, line 29 (emphasis added).

The specification of Jagannathan, moreover, details the purpose of each line of pseudo-code, reciting, for example, that in line 7 "... the [exception] dispatcher sends the exception to the target thread (line 7). Sending a thread a signal is equivalent to interrupting the thread and pushing … the signal handler … onto the thread's stack, and resuming the thread which causes the signal handler to be executed." *See,* Jagannathan, col 25, lines 22 – 26. Therefore, dispatching an exception to a thread necessarily includes sending the signal handler, and executing that signal handler (i.e., executing processor instructions).

The Office Action incorrectly states that the signal handler executes in response to delivered events. *See,* Office Action mailed April 18, 2007, pp. 3, ¶ 1. To the contrary, the signal handler is an integral aspect of exception delivery in that publication — e.g., as discussed above, the signal handler is pushed onto the thread's stack and executed. *See* Jagannathan, col. 25, lines 22 – 26. In order to have an exception dispatcher without instruction execution

according to Jagannathan, the signal handler would have to be removed, and there is no reason to believe that it would function without the signal handler. In sum, Jagannathan does not remedy the deficiencies of Brown.

Furthermore, as pointed out by Applicants in their submission dated January 29, 2007, there is no evidence of record suggesting that Brown and Jagannathan might be combined — nor that such a combination would be functional. As discussed above, Brown is purportedly directed towards processor improvements in the physical hardware of a microprocessor. Accordingly, the delivery mechanism purportedly disclosed in that publication is implemented at the hardware level. *See* Brown, col. 14, lines 1 – 17 and col. 7, lines 24 – 48. Jagannathan, on the other hand, is directed towards features which are several abstraction layers above the hardware level, and the exception dispatcher is implemented within the operating system. Those of ordinary skill in the art would not attempt to combine features of a delivery mechanism implemented at the hardware layer with a delivery mechanism implemented at the operating system layer.

In view of the foregoing, it is evident that the combination of Brown and Jagannathan fails to teach, suggest or otherwise render unpatentable the subject matter of claim 1. The same is true for claims 2 – 6 which depend from claim 1 and recite further limitations thereon.

Independent Claims 17, 28, 32, 35, 41, 47, 57, 61

Independent claims 17, 28, 32, 35, 41, 47, 57, 61 recite *inter alia* an event delivery mechanism that is in communication coupling with a plurality of processing units (or "virtual processing units" in claims 17, 32, 41, 47 and 61) and that delivers events to respective threads with which those events are associated without execution of instructions by said processing units (or "virtual processing units" in claims 17, 32, 41, 47 and 61). For at least the reasons discussed above, Brown and Jagannathan, either individually or in combination, fail to teach or suggest an event delivery mechanism with such limitations — and, thus, fail to render obvious the subject matter of claims 17, 28, 32, 35, 41, 47, 57, 61.

The same is true for claims 8 – 14, which depend from claim 7 and recite further limitations thereon; claims 18 – 23, which depend from claim 17 and recite further limitations thereon; claims 29 – 31, which depend from claim 28 and recite further limitations thereon;

claims 33 and 34, which depend from claim 32 and recite further limitations thereon; claims 36, 37, 39 and 40, which depend from claim 35 and recite further limitations thereon; claims 42 – 46, which depend from claim 41 and recite further limitations thereon; claims 48 – 56, which depend from claim 47 and recite further limitations thereon; claims 58 – 60, which depend from claim 57 and recite further limitations thereon; and claims 62 and 63, which depend from claim 61 and recite further limitations thereon.

IV. Rejection of Claims 4, 24 – 26, 38 and 53 – 55 Under 35 U.S.C. § 103

Dependent claims 4, 24 – 26, 38 and 53 – 55 stand rejected as allegedly unpatentable over Brown in view of Jagannathan, and further in view of Eggers (Eggers et al. "Simultaneous Multithreading: A Platform for Next-Generation Processors," IEEE, 1997; pages 12 – 29).

As discussed above, Brown and Jagannathan, both individually or in combination, fail to teach or suggest *inter alia* the event delivery mechanism of the claimed invention, as described above. Eggers does not teach or suggest a delivery mechanism with such limitations. The Examiner does not assert otherwise.

Accordingly, dependent claims 4, 24 – 26, 38 and 53 – 55 stand free and clear of the combined teachings of Brown, Jagannathan and Eggers. The 35 U.S.C. § 103 rejection should, therefore, be withdrawn.

V. Rejection of Claim 15 Under 35 U.S.C. § 103

Dependent claim 15 stands rejected as allegedly unpatentable over Brown in view of Jagannathan, and further in view of Gosior et al. (US 2003/0120896 A1; hereinafter "Gosior").

As discussed above, Brown and Jagannathan, both individually or in combination, fail to teach or suggest *inter alia* the event delivery mechanism of claim 7, as described above. Gosior does not teach or suggest such a mechanism – nor does the Examiner contend otherwise.

Accordingly, claim 15 stands free and clear of the combined teachings of Brown, Jagannathan and Gosior. The 35 U.S.C. § 103 rejection should, therefore, be withdrawn.

VI. Conclusion

As discussed above, the claimed subject matter is free and clear of the art and is otherwise in condition for allowance. The Examiner is encouraged to telephone the undersigned attorney for Applicants if such communication will expedite prosecution of this application.

Respectfully submitted,

Dated: August 13, 2007

/David J. Powsner/
David J. Powsner
Registration No.: 31,868

NUTTER MCCLENNEN & FISH LLP
World Trade Center West
155 Seaport Boulevard
Boston, Massachusetts 02210-2604
(617) 439-2000
(617) 310-9000 (Fax)
Attorney for Applicant